US 20100137045A2

(19) **United States**

(12) **Patent Application Publication**

Angelopoulos

(10) **Pub. No.: US 2010/0137045 A2**

(43) **Pub. Date: Jun. 3, 2010**

REPUBLICATION

(54) **METHOD AND SYSTEM FOR INCREASED REALISM IN VIDEO GAMES**

(75) Inventor: **Athanasios Angelopoulos**, San Diego, CA (US)

Correspondence Address:
Workman Nydegger
1000 Eagle Gate Tower
60 East South Temple
Salt Lake City, UT 84111 (UNITED STATES)

(73) Assignee: **White Knuckle Gaming, LLC**, Bountiful, UT (US)

(21) Appl. No.: **10/266,795**

(22) Filed: **Oct. 8, 2002**

Prior Publication Data

(65) US 2004/0067788 A1 Apr. 8, 2004

Publication Classification

(51) **Int. Cl.**
A63F 13/00 (2006.01)

(52) **U.S. Cl.** **463/1**

(57) **ABSTRACT**

A system and method for updating parameters of a video game is provided. As events occur in the real world that may influence a game attribute, a parameter is recorded on a server. A video game player may connect to the server and download the parameter. The parameter is stored by the video game and changes an attribute of the video game. Thus, for example, the new performance characteristics of the rising star may be recorded on the server as improved performance parameters. When these parameters are downloaded by the video game player, the video game incorporates the star-like qualities of the rising star.

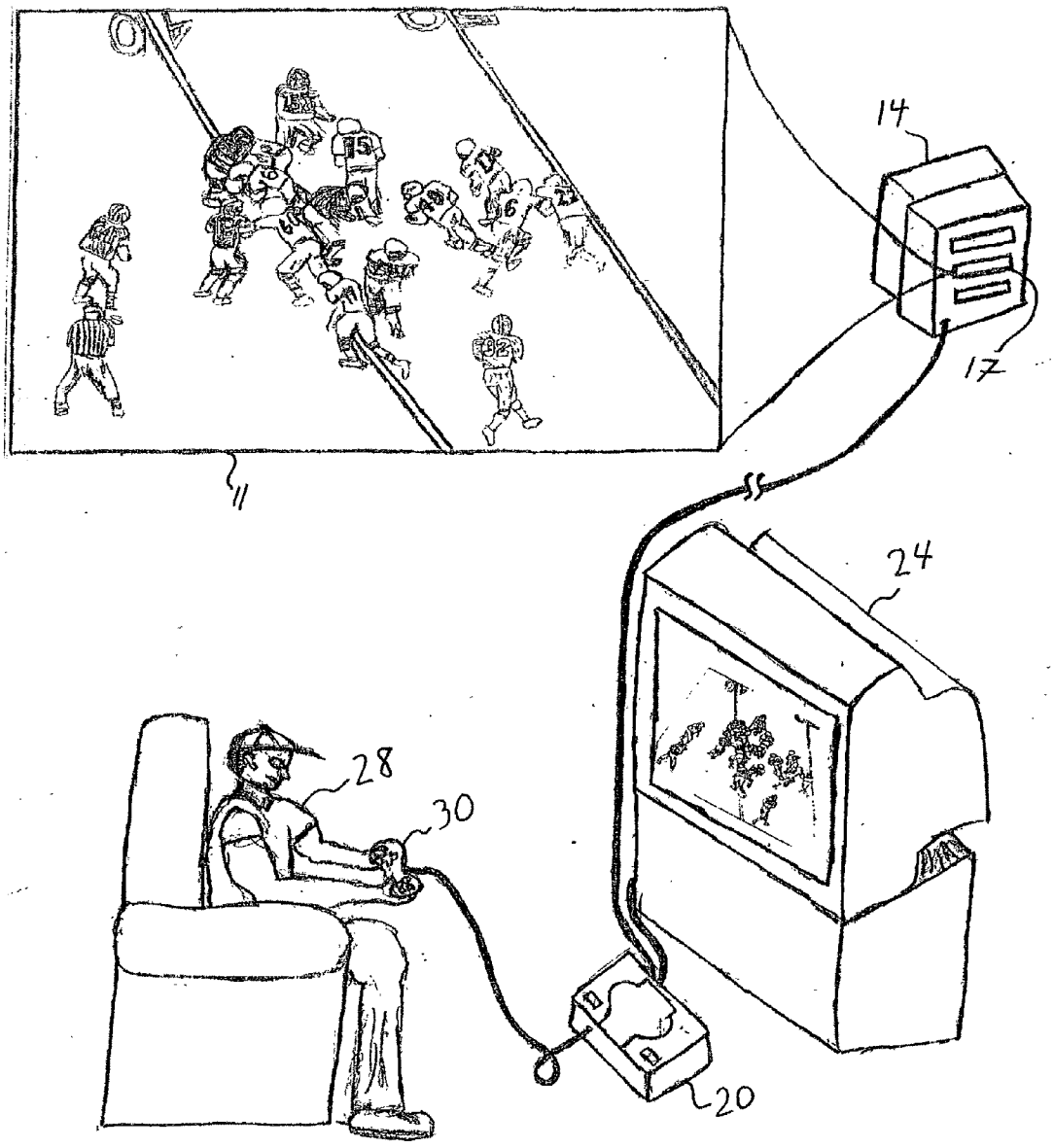


Fig. 1

Fig. 2

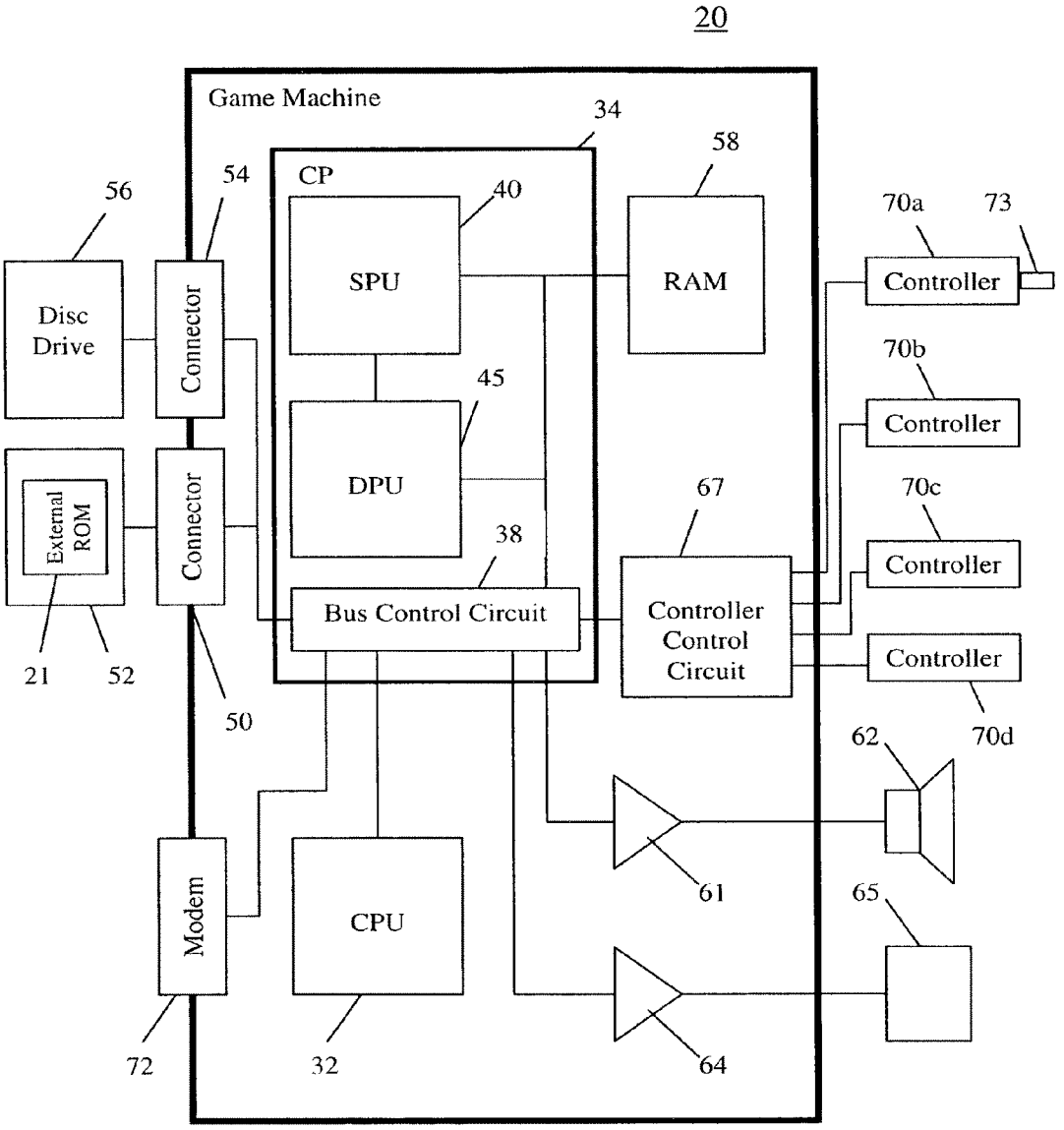

20
Game Machine
34
58
CP
54
56
40
70a
73
SPU
RAM
Controller
Disc Drive
Connector
70b
45
Controller
DPU
67
70c
External ROM
Connector
38
Controller
Bus Control Circuit
Controller Control Circuit
Controller
21
52
50
70d
62
61
65
Modem
CPU
72
32
64

Fig. 3

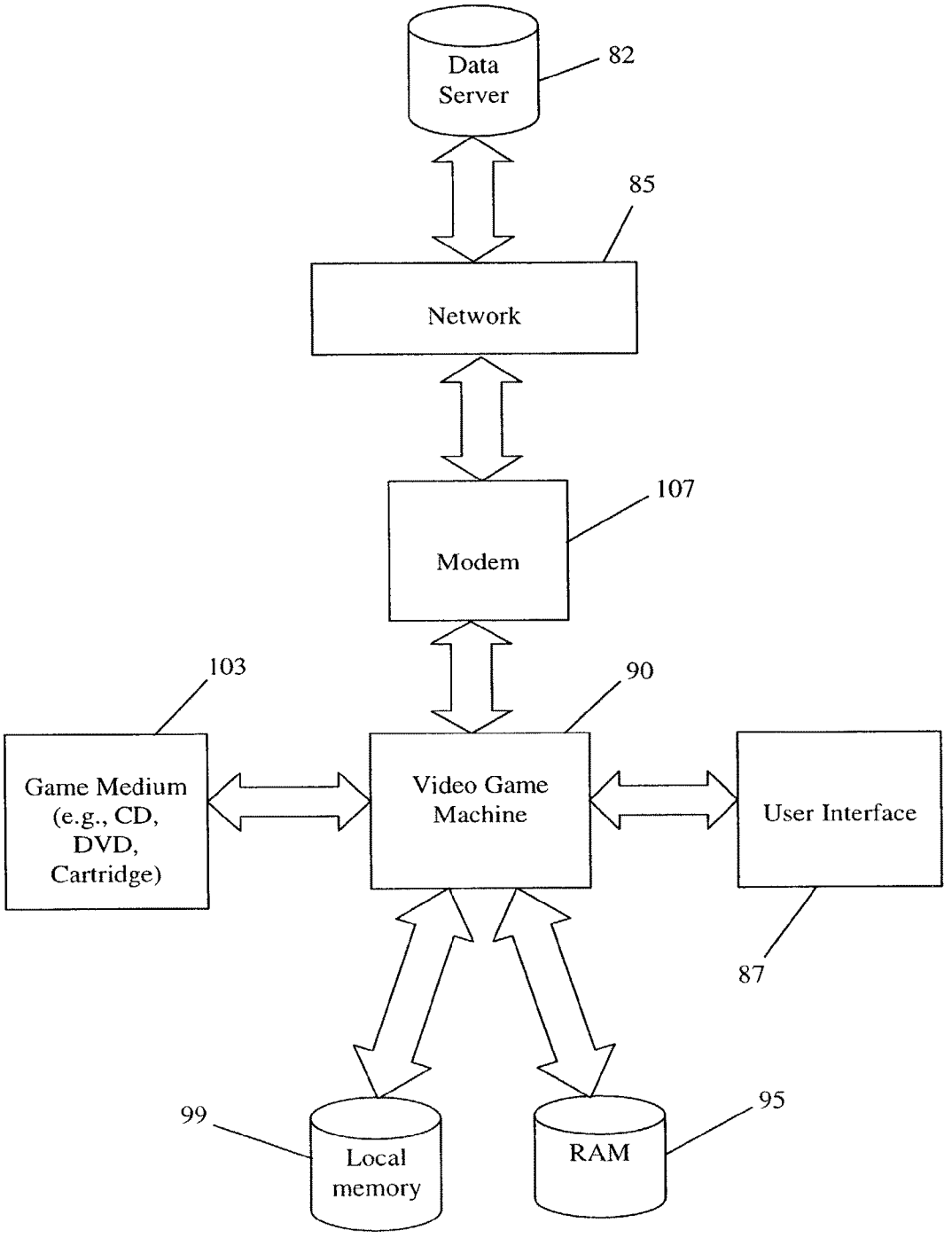

Data Server
82
85
Network
107
Modem
103
90
Game Medium (e.g., CD, DVD, Cartridge)
Video Game Machine
User Interface
87
99
Local memory
RAM
95

Fig. 4

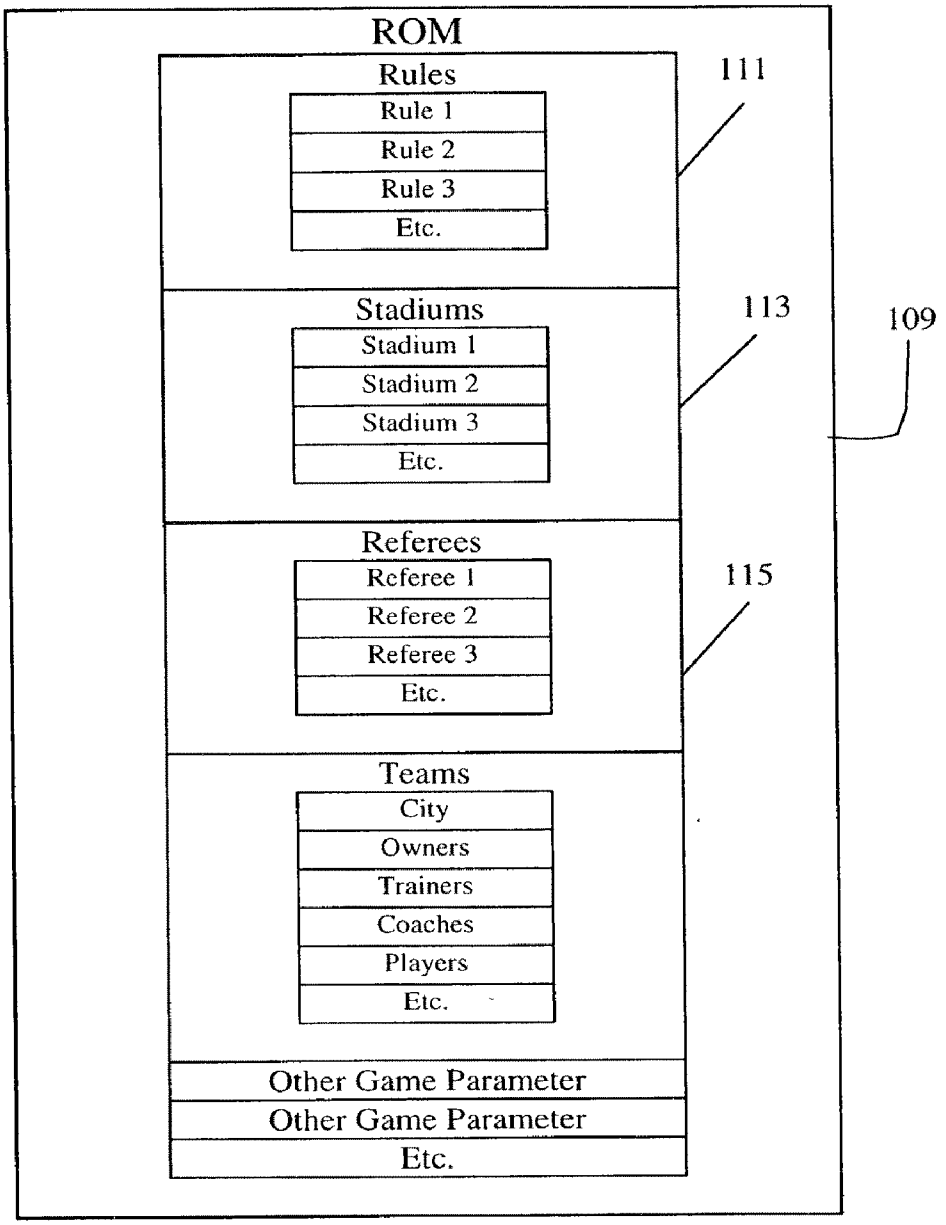

ROM
Rules
Rule 1
Rule 2
Rule 3
Etc.
111
Stadiums
Stadium 1
Stadium 2
Stadium 3
Etc.
113
109
Referees
Referee 1
Referee 2
Referee 3
Etc.
115
Teams
City
Owners
Trainers
Coaches
Players
Etc.
Other Game Parameter
Other Game Parameter
Etc.

Fig. 5

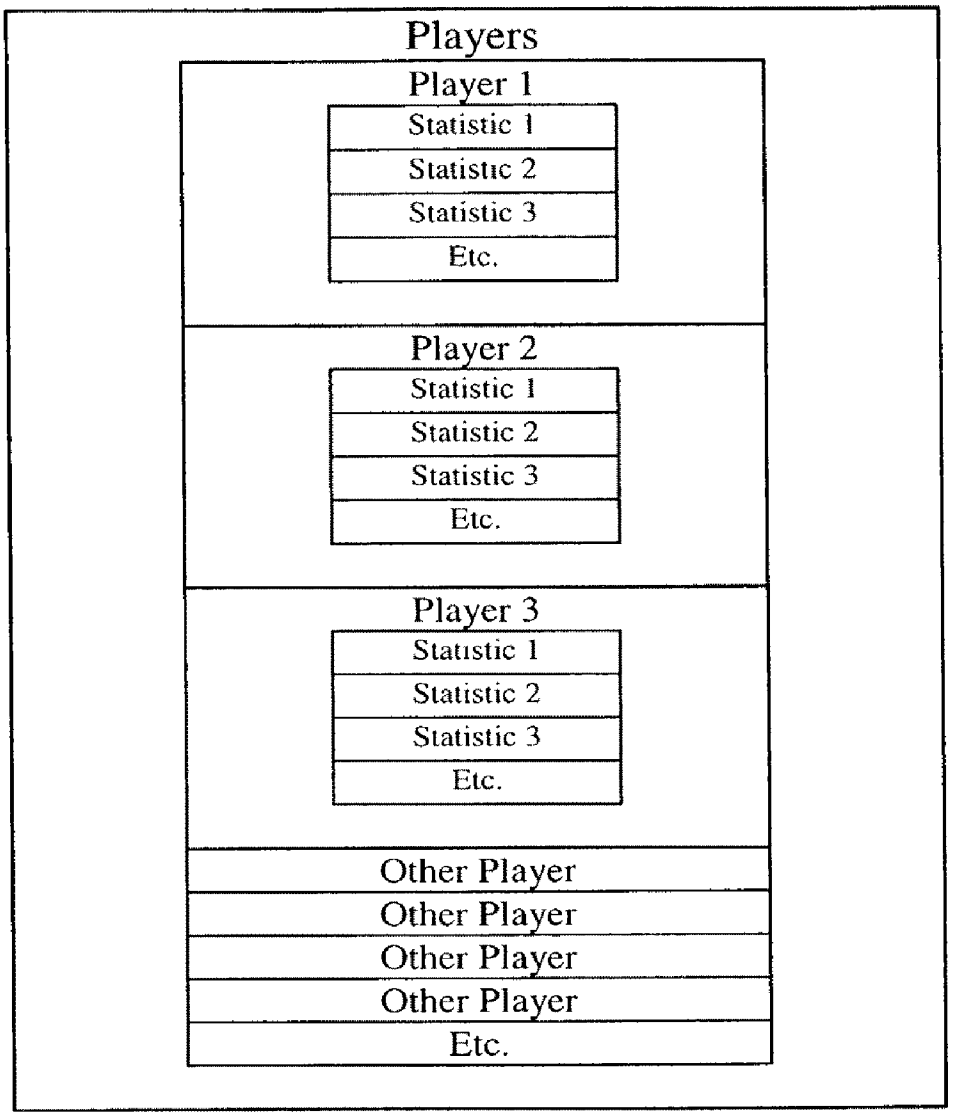

Players
Player 1
Statistic 1
Statistic 2
Statistic 3
Etc.
Player 2
Statistic 1
Statistic 2
Statistic 3
Etc.
Player 3
Statistic 1
Statistic 2
Statistic 3
Etc.
Other Player
Other Player
Other Player
Other Player
Etc.

Fig. 6

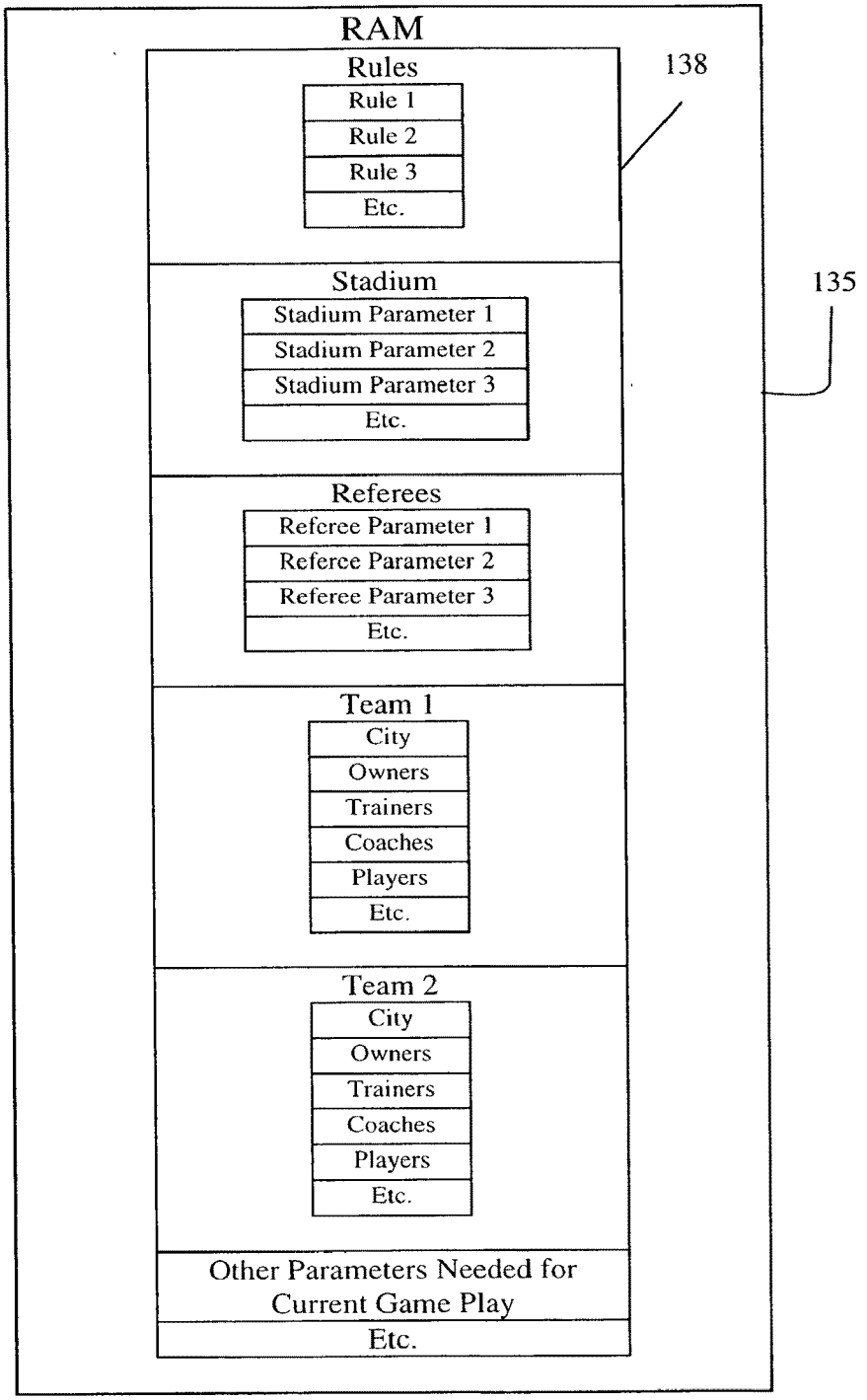

RAM
Rules
Rule 1
Rule 2
Rule 3
Etc.
138
Stadium
Stadium Parameter 1
Stadium Parameter 2
Stadium Parameter 3
Etc.
135
Referees
Referee Parameter 1
Refere Parameter 2
Referee Parameter 3
Etc.
Team 1
City
Owners
Trainers
Coaches
Players
Etc.
Team 2
City
Owners
Trainers
Coaches
Players
Etc.
Other Parameters Needed for Current Game Play
Etc.

Fig. 7

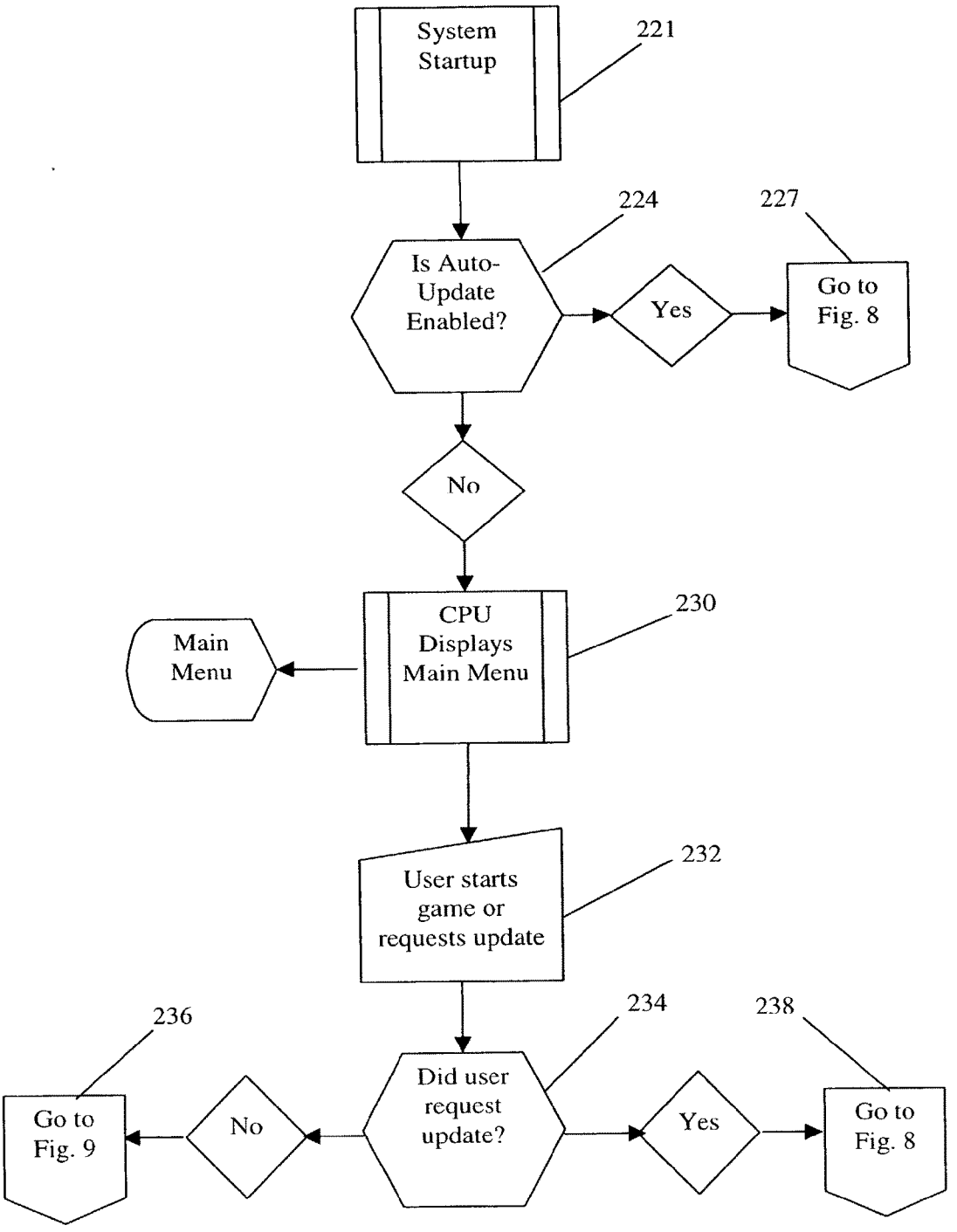

System Startup
221
224
227
Is Auto-Update Enabled?
Yes
Go to Fig. 8
No
CPU Displays Main Menu
230
Main Menu
User starts game or requests update
232
236
234
238
Go to Fig. 9
No
Did user request update?
Yes
Go to Fig. 8

Fig. 8

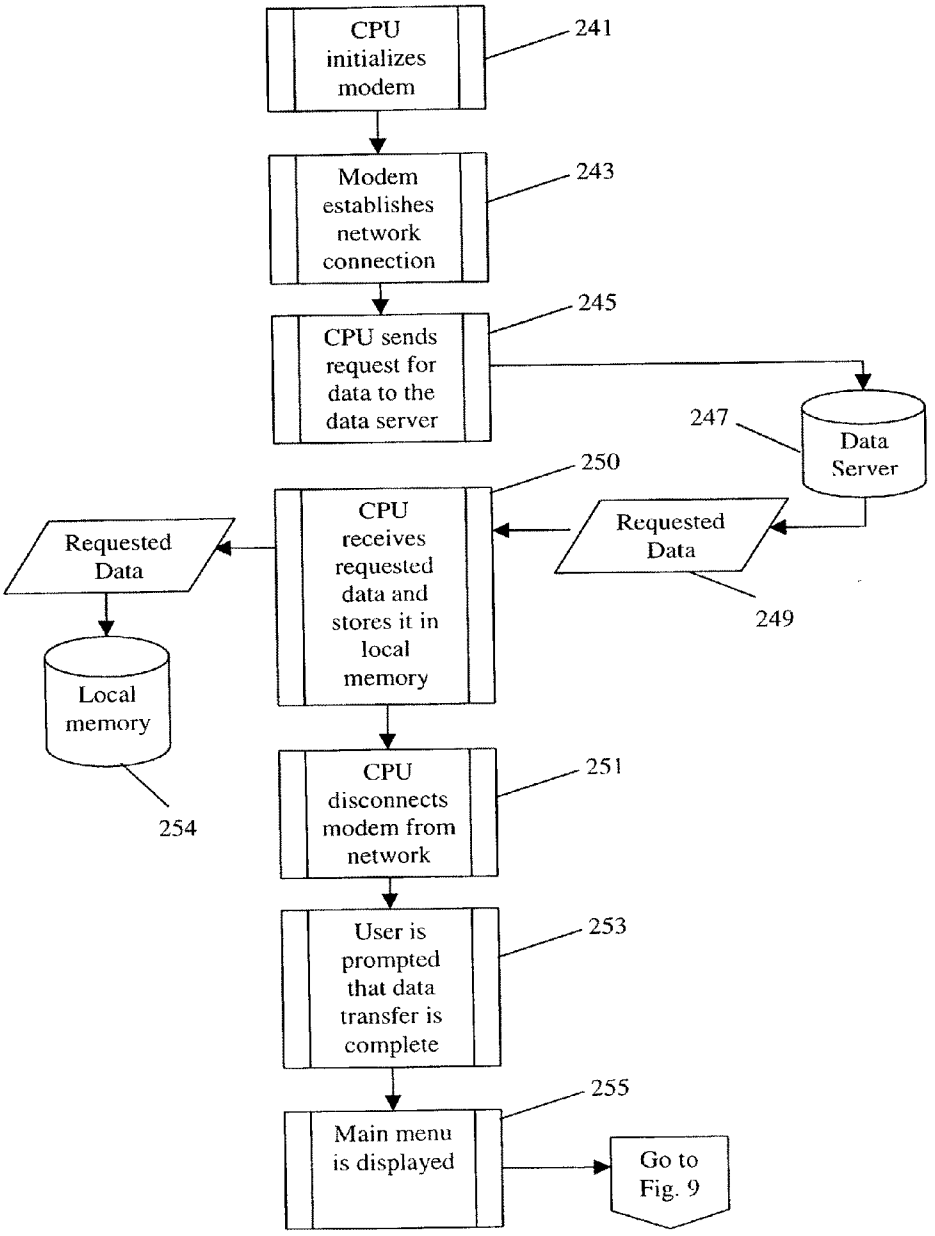

CPU initializes modem
241
Modem establishes network connection
243
CPU sends request for data to the data server
245
247
Data Server
250
CPU receives requested data and stores it in local memory
Requested Data
Requested Data
249
Local memory
254
CPU disconnects modem from network
251
User is prompted that data transfer is complete
253
Main menu is displayed
255
Go to Fig. 9

Fig. 9

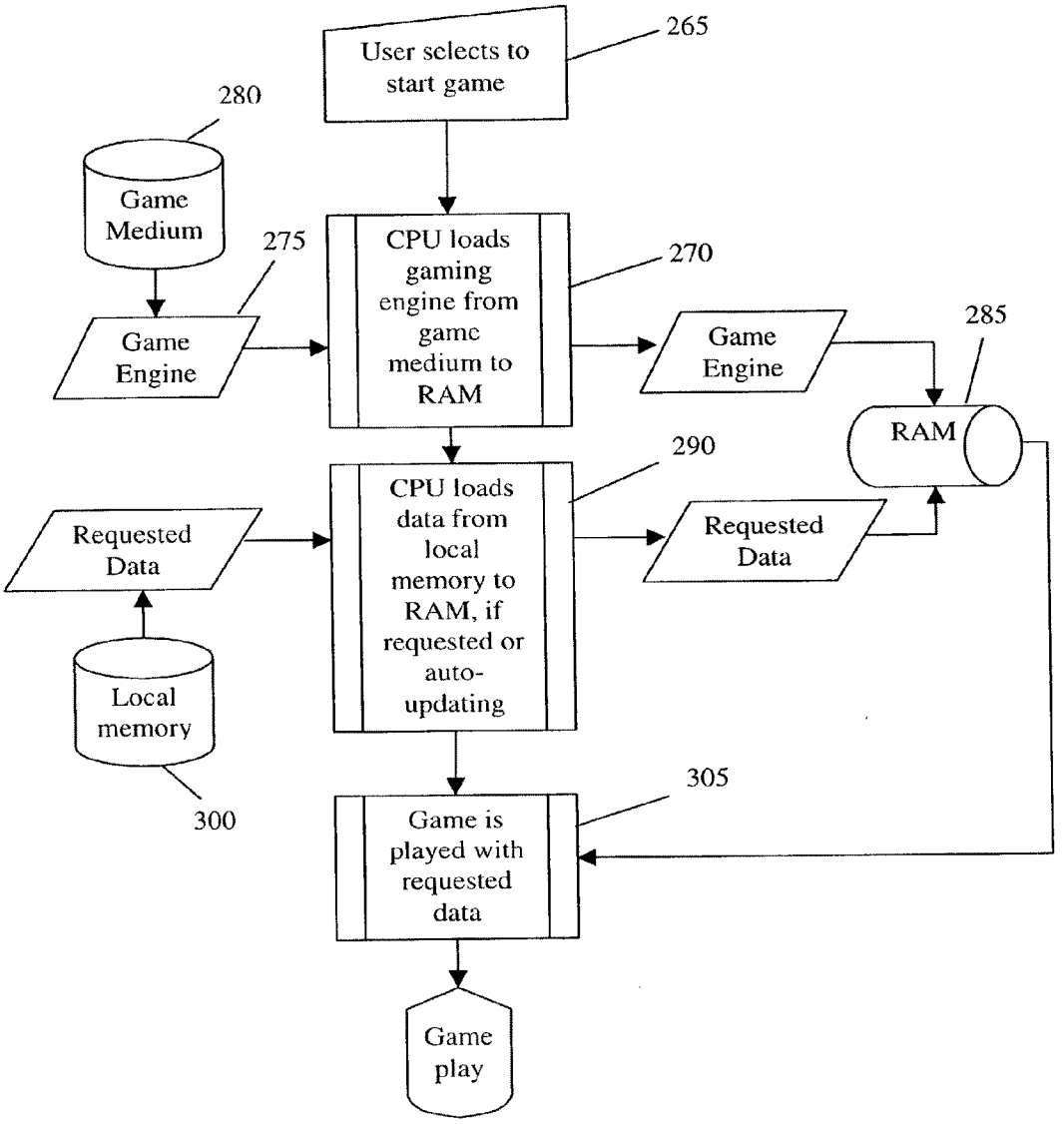

User selects to start game
265
280
Game Medium
275
Game Engine
CPU loads gaming engine from game medium to RAM
270
Game Engine
285
RAM
290
CPU loads data from local memory to RAM, if requested or auto-updating
Requested Data
Requested Data
Local memory
300
305
Game is played with requested data
Game play

Fig. 10

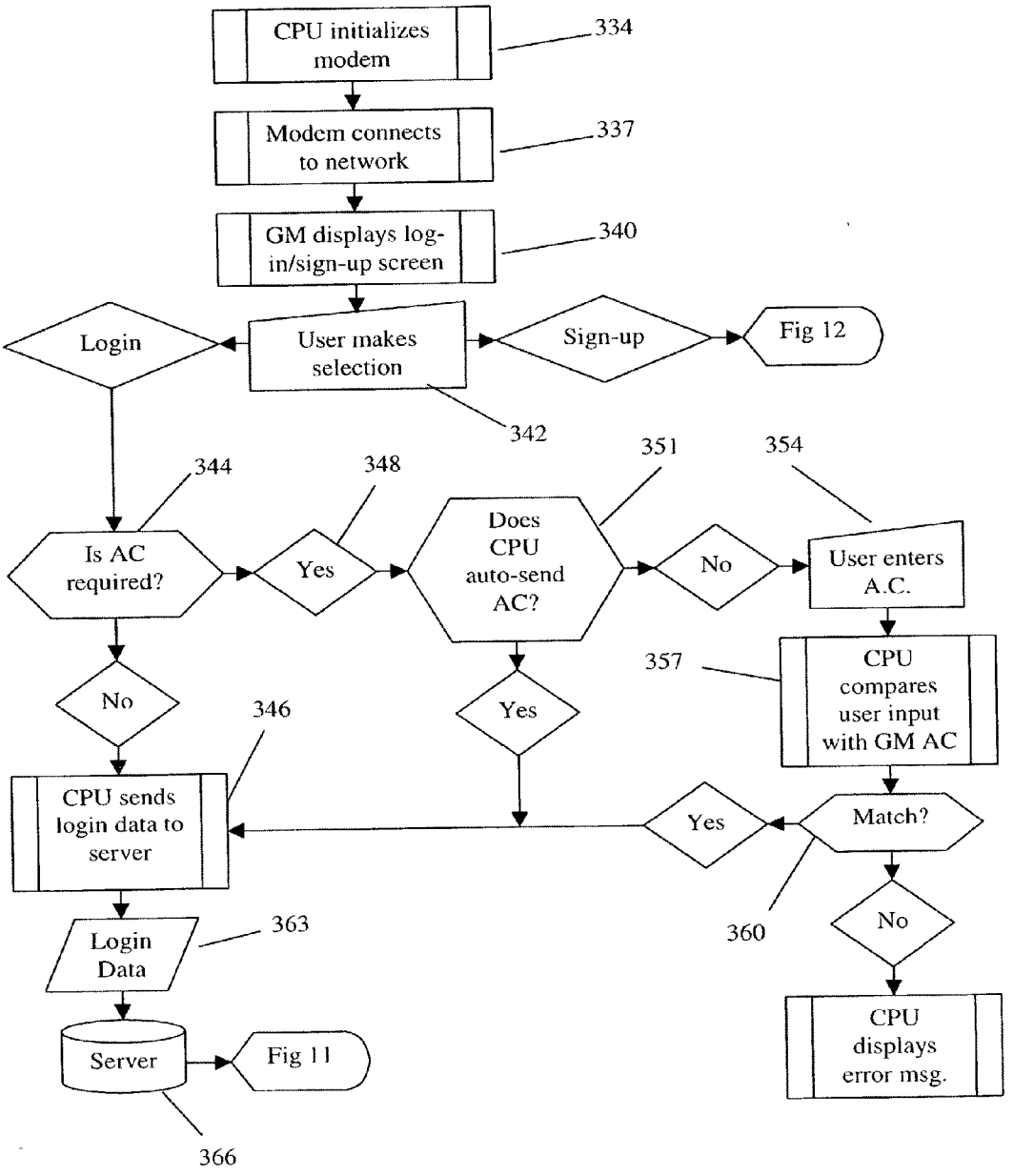

CPU initializes modem
334
Modem connects to network
337
GM displays log-in/sign-up screen
340
Login
User makes selection
Sign-up
Fig 12
342
344
348
351
354
Is AC required?
Yes
Does CPU auto-send AC?
No
User enters A.C.
No
Yes
357
CPU compares user input with GM AC
346
CPU sends login data to server
Yes
Match?
360
No
363
Login Data
Server
Fig 11
CPU displays error msg.
366

Fig 11

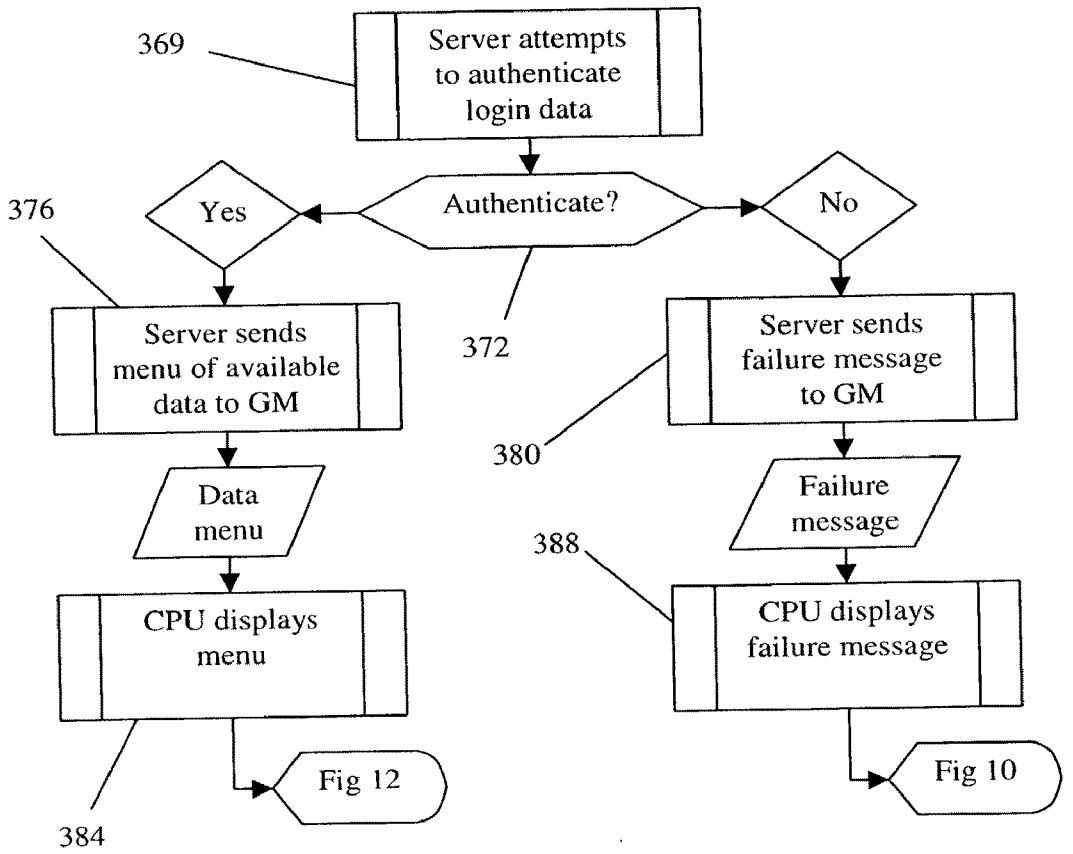

369
Server attempts to authenticate login data
Authenticate?
Yes
No
376
372
Server sends menu of available data to GM
Server sends failure message to GM
380
Data menu
Failure message
388
CPU displays menu
CPU displays failure message
Fig 12
Fig 10
384

Fig 12

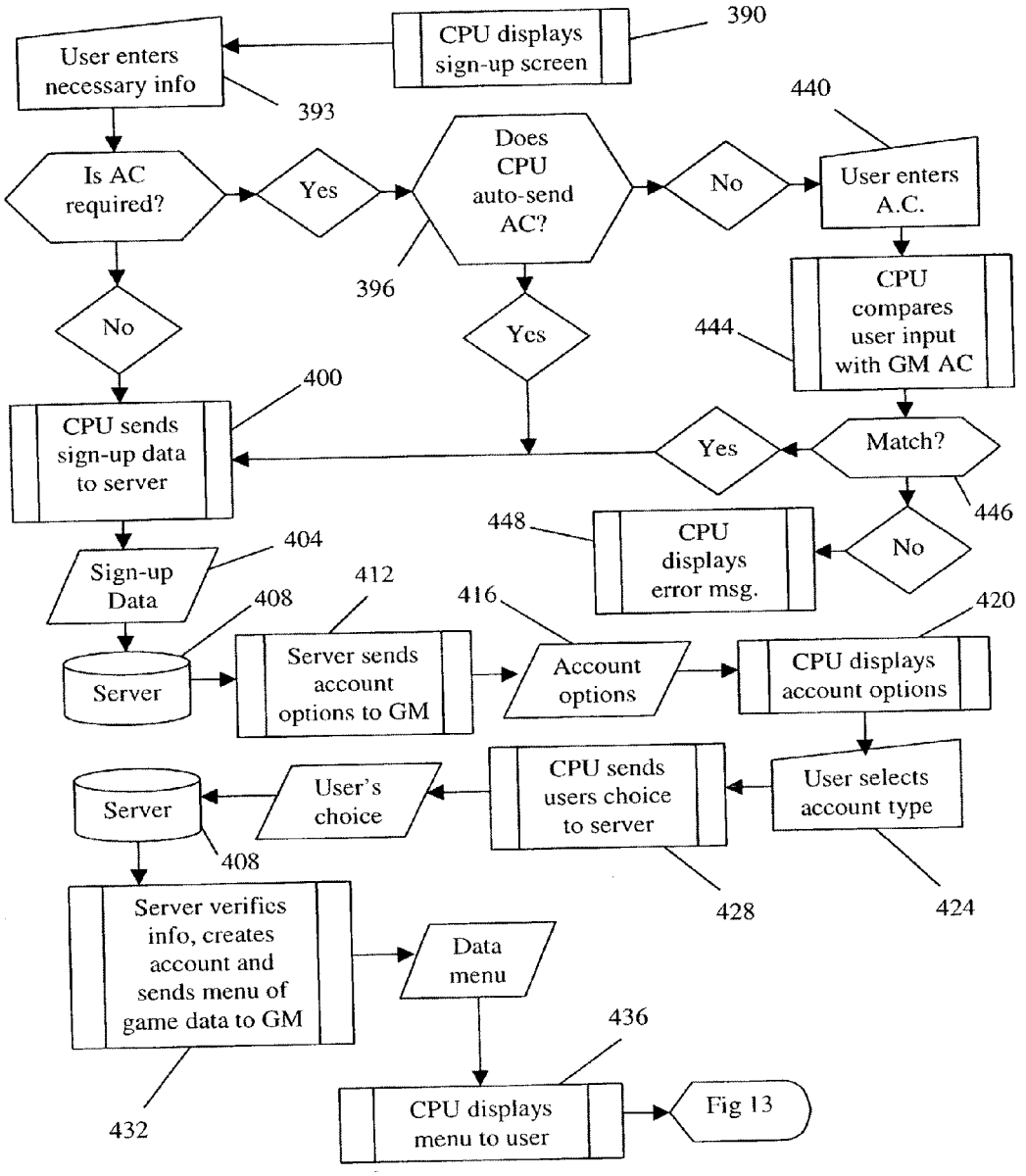

CPU displays sign-up screen
390
User enters necessary info
393
Is AC required?
Yes
Does CPU auto-send AC?
396
No
User enters A.C.
440
No
Yes
CPU compares user input with GM AC
444
CPU sends sign-up data to server
400
Yes
Match?
446
No
448
CPU displays error msg.
Sign-up Data
404
Server
408
412
Server sends account options to GM
416
Account options
CPU displays account options
420
User selects account type
CPU sends users choice to server
428
User's choice
Server
408
424
Server verifies info, creates account and sends menu of game data to GM
432
Data menu
436
CPU displays menu to user
Fig 13

Fig 13

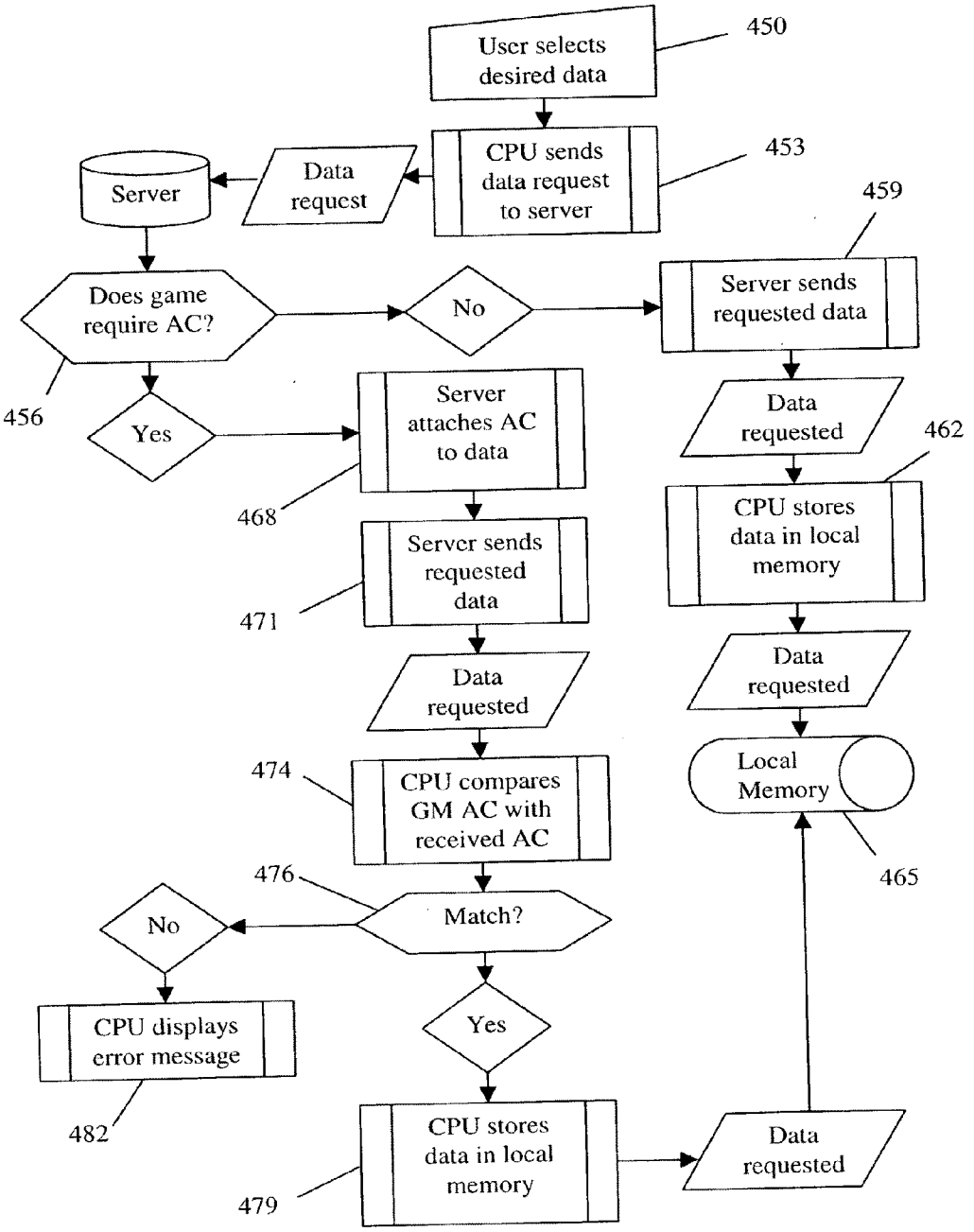
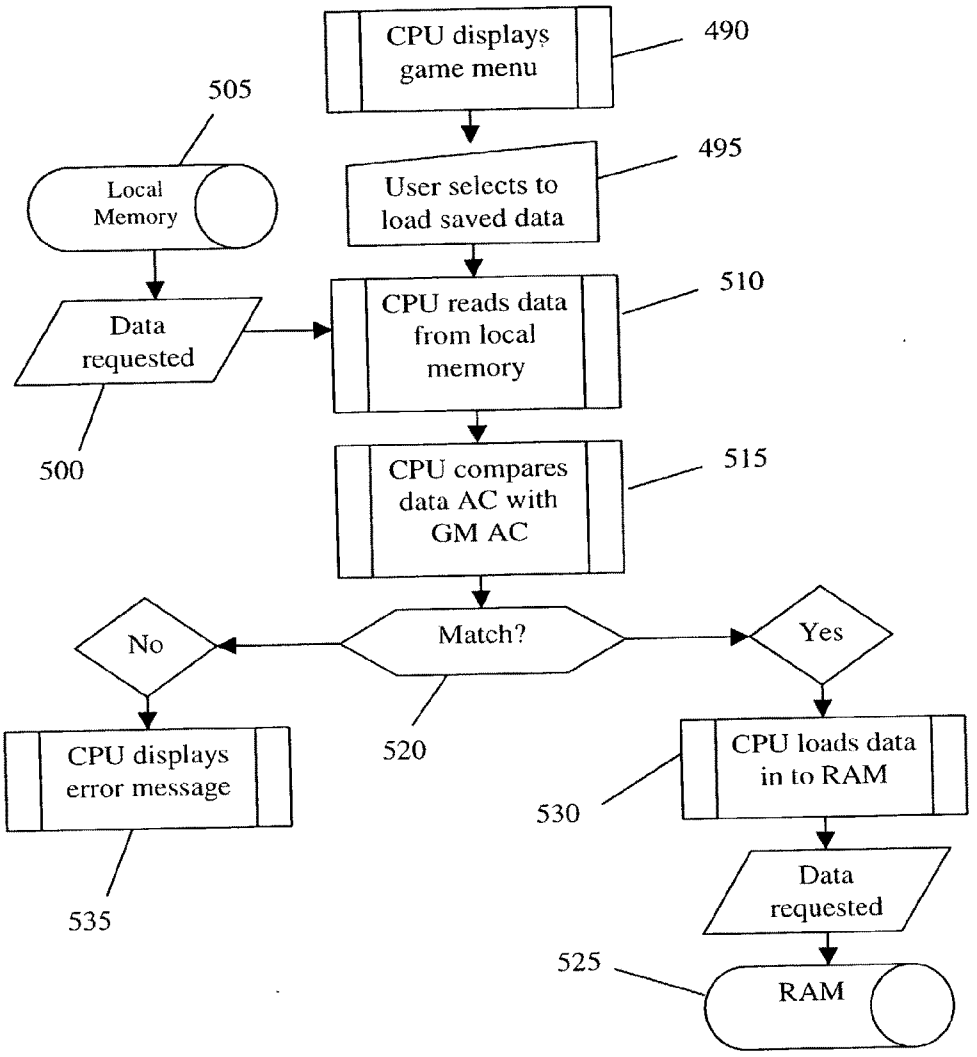

User selects desired data
450
CPU sends data request to server
453
Data request
Server
Does game require AC?
No
459
Server sends requested data
Data requested
462
CPU stores data in local memory
Data requested
Local Memory
465
456
Yes
Server attaches AC to data
468
Server sends requested data
471
Data requested
474
CPU compares GM AC with received AC
476
Match?
No
CPU displays error message
482
Yes
CPU stores data in local memory
479
Data requested

Fig 14


CPU displays game menu
490
505
Local Memory
User selects to load saved data
495
Data requested
CPU reads data from local memory
510
500
CPU compares data AC with GM AC
515
No
Match?
Yes
CPU displays error message
520
CPU loads data in to RAM
530
Data requested
525
RAM
535

METHOD AND SYSTEM FOR INCREASED REALISM IN VIDEO GAMES

BACKGROUND

[0001] Many video games attempt to recreate realistic situations. For example, video games simulating sports games and matches are popular. They commonly even show actual leagues with correct teams. Home towns may be shown, with realistic stadiums. Athletes' uniforms represent professional athletes' actual uniforms. Specific athletes are represented. For example, an image of Shaquile O'Neil may be used as one of the players on the Los Angeles Lakers basketball team. Kobe Bryant and all of Shaquile's actual teammates may be on the team with him.

[0002] But the realism goes far beyond just images. The video game characters have characteristics approximating their real life counterparts. For example, the video game Shaquile may be very good at dunking the basketball, but horrible at shooting freethrows. Parameters are stored on the video game medium that cause the different characters to replicate their real life counterparts' performance.

[0003] Video game producers typically produce professional sports video games once per year. The performance parameters and visual aspects are typically specified based on the most recent year. For example, Barry Bonds of the San Francisco Giants baseball team had a homerun average of (73 homeruns)/(476 at bats)=15.3 percent during the 2001 season. This average can be used to set a video game parameter so that the video game character Barry Bonds has a similar tendency to hit homeruns. A problem with this system is that the performance parameters and visual aspects of a video game are fixed at the time the video game is produced.

[0004] The players of this type of game continually demand increased realism in such games. For example, more and more parameters are added to distinguish one character from another and to more accurately simulate the performance of real life athletes. Also, the realism of the visual aspects of players, teams and stadiums is continually increased. For example, the ivy on the outfield wall of Wrigley Field in Chicago is may be shown in a baseball video game.

[0005] The problem remains that visual aspects and performance characteristics of real life sports players, teams and environments change throughout the year.

SUMMARY

[0006] Sports video games attempt to achieve realism. Particularly, sports video games frequently attempt to emulate professional sport details. Actual professional athletes are shown, frequently including such details as jersey numbers, physical stature and even facial features. Performance characteristics are emulated. For example, in a baseball video game, an individual professional athlete's batting average may be used to influence a video game character's batting performance. Frequently actual professional teams are provided for video game play. For example, in a basketball video game, players can play the Los Angeles Lakers versus the Chicago Bulls, complete with rosters, players and even a stadium. Video game players are demanding more and more realism like this.

[0007] A major problem with this system is that professional sports change in many ways. Athletes' performances change over the course of a season, or even a day. Additionally, athletes may be traded from team to team, or become injured. Stadia may change. Sports announcers may change. This leaves a video game behind, in an unrealistic state. For example, if a new star rises on the professional sports scene, during the middle of a season, that new star's incredible performance will not be duplicated by the video game. Video game players are stuck playing the game with the star having his pre-star characteristics. They must wait for the video game producers to produce a new video game incorporating the new star-like performance of the rising star. This is unsatisfying for video game players. They want to play a video game that matches the professional sport performance characteristics, look and feel.

[0008] Accordingly, a system and method for updating parameters of a video game is provided. As events occur in the real world that may influence a game attribute, a parameter is recorded on a server. A video game player may connect to the server and download the parameter. The parameter is stored by the video game and changes an attribute of the video game. Thus, for example, the new performance characteristics of the rising star may be recorded on the server as improved performance parameters. When these parameters are downloaded by the video game player, the video game incorporates the star-like qualities of the rising star.

[0009] This quantifiably increases realism. Video game players can download up-to-the-minute statistics and visual aspects for incorporation into their video games. Instead of playing with last season's teams, video garners get a simulation of the fresh new events of the week, day, or hour.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. **1** is an external view showing a structure of a video game system and a historical event.

[0011] FIG. **2** is a high level block diagram of a video game system.

[0012] FIG. **3** is a block diagram of a video game system.

[0013] FIG. **4** is a block diagram of a game medium.

[0014] FIG. **5** is a block diagram of the players stored in ROM.

[0015] FIG. **6** is a block diagram of a RAM.

[0016] FIG. **7** is a flow chart for choosing whether to update a video game.

[0017] FIG. **8** is a flow chart for updating a video game.

[0018] FIG. **9** is a flow chart for loading data into RAM and playing a video game.

[0019] FIG. **10** is a block diagram showing a portion of a flow diagram for authenticating a user of a video game.

[0020] FIG. **11** shows a portion of a process for authenticating a video game user, continued from FIG. **10**.

[0021] FIG. **12** is a flow diagram showing a process for signing up a new user of a video game with downloadable statistics.

[0022] FIG. **13** is a flow diagram showing a download process of a new game parameter.

[0023] FIG. **14** is a flow diagram showing game play in a case in which an AC is required.

DETAILED DESCRIPTION OF SEVERAL EMBODIMENTS

[0024] FIG. **1** is an is an external view of a video game system and a historical event. An historical event **11** takes place. For example, the event may be a World Cup soccer game, or an American football game. In a more specific example, the event may be the occurrence of a new yearly completion percentage, or other performance statistic, of a specific quarterback in a specific professional American football game.

[0025] A datum **17** relating to the historical event **11** is recorded on a computer readable medium known as a network server **14**. A game machine **20** is coupled to the network server **14**. The game machine **20** downloads the datum **17**, which may, for example, represent a quarterback's completion percentage, a basketball player's freethrow percentage, or any other outcome of a realistic event. The datum **17** is then used to play a video game, which is displayed on a monitor **24**. A user **28** plays the video game using controller **30**. In this way, the user gets to play a video game the more accurately simulates a the real world.

[0026] For example, a datum representing a quarterback's new completion percentage is loaded into memory (not shown) in the game machine **20**. Then, the user **28** can play an American football video game with current statistics. This makes the video game more enjoyable to the user, adding increased realism. As real life professional players' performances change, users can still play with statistically accurate players.

[0027] FIG. **2** is a block diagram of a video game system. The video game machine **20** incorporates a central processing unit (CPU) **32** and coprocessor (CP) **34**. The CP **34** includes a bus control circuit **38** for controlling buses, a signal processing unit (SPU) **40** for performing polygon coordinate transformation, shading treatment, etc., a display processing unit (DPU) **45** for rasterizing polygon data into an image to be displayed and converting the data into a data from (dot data) stored on a froame memory. The CP **34** is coupled to a cartridge connector **50** for detachably mounting with a ROM cartridge **52**, a disc drive connector **54** for detachably mounting with a disc drive **56**, and a RAM **58**. Also, the CP **34** is connected with an audio signal generating circuit **61** for outputting a sound signal to an audio output device **62**, processed by the CPU **32**, and an image signal generating circuit **64** for outputting an image signal to a display **65**. Further, the CP **34** is coupled with a controller control circuit **67** for serially transferring operational data for one or a plurality of controllers **70**A-**70**D and data for a RAM cartridge **73** for extension.

[0028] A modem **72** is coupled to the bus control circuit **38**. The modem **72** is also preferably coupled to the internet (not shown). As discussed with reference to FIG. **1**, a server coupled to the internet stores performance, visual image and audio parameters. The modem **72** couples to the server and downloads a parameter for play in the video game. The CPU **11** sends a signal through the bus control circuit to initialize the modem **72** and control the modem **72** to download the parameter. Preferably, the parameter is stored in RAM **58** for play of the video game, as discussed below. It will be understood by those of skill in the art that other configurations of video game machines are possible. For example, the CPU II and the CP **34**, or portions of the CP **34**, may be incorporated as one component. Also, for example, RAM **58**, may be divided into more than one memory element, or other types of memory elements may be included.

[0029] Referring now to FIG. **3**, a block diagram of a video game system **79** is shown. A data server **82** is coupled to a network **85**. The network may be, for example, what is commonly referred to as the internet, or, alternatively, the network **85** may be another type of network, such as a local area network (LAN). The data server **82** stores information for use in video games. Advantageously, new statistical results or parameters can be stored on the data server **82**. For example, when a real professional athlete changes a performance rating, the changed performance rating can be stored on the data server **82**. For example, when a professional baseball player changes his batting average, the changed batting average can be stored on the data server **82**. Or, as a second example, if a professional American football quarterback changes his completion percentage the changed completion percentage can be stored on the data server **82**. This allows a video game player to play a video game with the new statistics. The changed statistic may be stored as the actual number that the statistic represents. For example, if a batting average goes from 0.275 to 0.278, then the number 0.278 may be stored on the data server. Or the new statistic may be stored on the data server as a parameter of a video game. The parameter can then be used to effect how the video game character interacts with the video game environment.

[0030] Referring again to FIG. **3**, the user (not shown) plays the video game by entering inputs through a user interface **87**. The user interface **87** may be, for example, a video game controller (not shown). The user interface **87** is coupled to a video game machine **90**, which may be a video game machine **20**, as shown with reference to FIG. **2**. The video game machine **90**, is coupled to a RAM **95** and a local memory **99**, preferably a read only memory (ROM). The RAM **95** is used for storing data and rules (described later with reference to FIG. **4**) from the game medium **103**. Advantageously, the parameter is also stored on the RAM. Thus, the video game is played with the changed statistic. Advantageously, a video game character, such as, for example, an American football quarterback, will play more realistically based on the video game character's real life performance, such as, for example, a completion percentage.

[0031] The video game machine is coupled to a modem **107** for coupling to the network **85**. The parameter is downloaded by the video game machine **90** from the data server **82** by means of the modem **107** and the network **85**. As will be understood by those of skill in the art, the modem **107** may be incorporated as part of the video game machine **90**, or a separate component.

[0032] FIG. **4** is a block diagram of a game medium, preferably a ROM **109**. The game medium may be, for example, a compact disc (CD), a digital video disc (DVD), or a cartridge. The ROM **109** includes many rules and parameters for the video game. Rules **111** are stored on the ROM **109**. The rules **111** are shown as Rule **1**, Rule **2**, Rule **3**, etc. The rules govern how the game is played and displayed on the user interface. Advantageously, the rules can be updated to match the rules of the real world.

[0033] There are also other parameters stored on the ROM **109**. For example, stadium parameters **113** may be stored. For

example, the game may display Wrigley Field for play by the Chicago Cubs and Fenway Park for play by the Boston Red Socks. For example, the ivy on the homerun wall of Wrigley Field is green and flourishing during certain parts of the year, such as from May to July. During April, August and September, the ivy is browner. The color of the ivy shown in a baseball video game showing Wrigley Field can change based on the time of the year that the video game is played.

[0034] As another example of how realism can be increased, new memorials to great players can be implemented in the video game stadiums. For example, in 2002, Ted Williams' number, "9", was displayed in the leftfield grass of Fenway Park after the start of the season. Baseball video games were already produced. So the video game fields did not have the number on the field, failing to duplicate real life. Now, the field can be changed to match real life after the game is produced, sold and played in the user's home.

[0035] Referring again to FIG. **4**, referees **115** may be stored. Referees' appearances and performance may be stored. For example, some referees in basketball call technical fouls frequently. Some referees rarely call technical fouls. Parameters can be stored regarding the individual referees that simulate these real life differences.

[0036] Team statistics and other parameters can also be stored, such as, for example, the city, owners, trainers, coaches, players, etc. For example, Mike Bibby of the Sacramento Kings, during the 2001-2002 regular season averaged 13.7 points and 12.3 shots attempted per game. In the playoffs of the same season, he averaged 20.2 points and 16.1 shot attempted per game. These improved playoff statistics can now be downloaded. A video game player can play a basketball video game with a Mike Bibby character incorporating the better performing real world Mike Bibby.

[0037] As another example, the 2000-2001 Chicago Bears' opponents scored an average of 22.2 points per game with an average of 114.2 rushing yards per game. The 2001-2002 Chicago Bears' opponents scored an average of 12.7 points per game with an average of 82.1 rushing yards per game. This change was not reflected in video games played during the 2001-2002 season. At that time, the newest video games reflected the statistics from the 2000-2001 season. Thus, the Chicago Bears on the video games played poorly, whereas the Chicago Bears on the real football field played very well. This dichotomy between video game and real life is frustrating to many video game players.

[0038] FIG. **5** is a block diagram of the players stored in ROM. Player **1**, Player **2**, Player **3** and several other players are shown. Under each player, several statistics are shown. For example, if the video game is basketball, Statistic **1** may be a player's field goal percentage. Statistic **2** may be a free throw percentage. Statistic **3** may be blocks per game. One skilled in the art will appreciate that the number and type of statistics stored will very with the game and with the way the video game designer choses to design the game.

[0039] FIG. **6** is a block diagram of a RAM **135** used to store game parameters for play of a video game. RAM **135** may be used to store game parameters as are RAM **58** and RAM **95**. Preferably, items are loaded from ROM **109** into RAM **135**. Preferably, all of the rules **111** are loaded into RAM at **138**. However, only one stadium is loaded from ROM to RAM. Also, only two teams, shown as Team **1** and Team **2** are loaded. As discussed with reference to FIGS. **1-3**, above, a new parameter is downloaded from the server and stored in RAM **135**. Preferably, the new parameter is stored in RAM in addition to the parameter as loaded from ROM.

[0040] Referring now to FIG. **7**, a flowchart is shown. In step **221**, the method starts. In step **224**, it is determined whether autoupdate is enabled. If autoupdate is enabled, the method continues by going to FIG. **8**, as shown at step **227**. If autoupdate is not enabled, the main menu is displayed at step **230**. In step **232**, a user chooses to start a game or request an update, at step **232**.

[0041] In step **234**, it is determined whether the user requested an update. In step **236**, the method continues by going to FIG. **9**, if the user did not choose to update the video game. If the user did choose to update the video game, the method continues by going to FIG. **8**, at step **238**.

[0042] Referring now to FIG. **8**, in step **241**, the CPU initializes the modem. In step **243**, the modem establishes a network connection. Next, in step **245**, the CPU sends a request for data to the data server. At **247** and **249**, the data server sends the requested data to the CPU. Advantageously, the requested data may be updated, or changed, statistics for playing a more realistic video game. For example, as mentioned above, the user may be requesting updated batting averages for game characters in a profession baseball video game. Or, to continue the example from above, the video game player may be requesting a newly established completion percentage for an American football quarterback. Preferably, the request is for all the new statistics resulting from a real game that has already been played, or from a plurality of games.

[0043] Referring again to FIG. **8**, at step **250**, the CPU receives the requested data and stores it in local memory **254**. In step **251**, the CPU disconnects the modem from the network. Next, at step **253**, the user is prompted that the data transfer is complete. The main menu is displayed at step **255**, and the method continues to FIG. **9** for play of the game.

[0044] In FIG. **9**, a flowchart is shown for the play of the video game. At step **265**, the game starts. Next, at step **270**, the CPU loads the game engine **275** from the game medium **280** into RAM **285**. Next, at step **290**, the CPU loads the requested data **295** from local memory **300** into RAM **285**. The requested data **295** is preferably the changed statistics requested by the user and stored in local memory as shown in FIG. **8**. This way, updated statistics, such as new player performance parameters resulting from current, real professional games can be used to play a more realistic video game, as shown at step **305**.

[0045] FIG. **10** is a block diagram showing a portion of a flow diagram for authenticating a user of a video game. At step **334**, a CPU initializes a modem. The modem connects to a network at step **337**. The game machine displays a login/sign-up screen at step **340**. The user selects either to sign up or login in at step **342**. If the user selects to login, the next step depends upon whether an authentication code (AC) is required, as shown at step **344**. If no AC is required, the CPU sends the login data to the server at step **346**. The flow then continues at FIG. **11**.

[0046] Alternatively, an AC may be required by the network, as shown at step **348**. This allows the network to verify that the user has a proper authentication code. For example,

the authentication code may be provided at sign-up (described with respect to FIG. **12**) by the network to the user. This way, the network administrator can verify that the user is a paying user. As another example, a new authentication code may be provided on a periodic basis, such as, for example, monthly. Thus, if a user is paying on a monthly basis, correct authentication is only available if the user's dues are current. As yet another example, the AC may be associated with a specific game machine by either the network or the CPU. This allows for security for preventing portability of AC's from one game machine to another.

[**0047**] Referring again to FIG. **10**, if authentication is required, it depends whether the AC is auto-sent or not, as shown at step **351**. If the AC is auto-sent, the process continues at step **346**. If the AC is not auto-sent, the user enters a user verification code, at step **354**. At step **357**, the CPU compares the user verification code to a game machine authentication code. If the user verification code matches the game machine authentication code, at step **360**, the process continues at step **346**. As stated in step **346**, the login data **363** is stored at a network server **366**. Next, the process continues at FIG. **11**.

[**0048**] FIG. **11** shows a portion of a process for authenticating a video game user, continued from FIG. **10**. At step **369**, the server attempts to authenticate the login data and any AC present. The AC may include a user verification code, to verify that the user is authorized to receive updated game parameters. Also, the AC may include a code identifying a specific game machine. This identifying code may, for example, be a serial number from the game machine. Alternatively, the identifying code may be a code assigned by the network administrator to the game machine at sign-up (described with respect to FIG. **12**). Further, the network administrator may verify that the identifying code, the verification code, and the user login data all match. The verification code may be a password.

[**0049**] At step **372**, the network checks whether the login data and any AC is authenticated. If the login data and any AC are authenticated, at step **376**, the server sends a menu of available data to the game machine. Then, at step **384**, the CPU displays the menu to the user and continues to FIG. **13**. If the login data and any AC are not authenticated at step **372**, the server sends a failure message to the CPU, at step **380**. The CPU displays the failure message to the user, at step **384**. Preferably, the CPU next displays the login/sign-up screen to the user again. The process returns to the beginning of FIG. **10**, at step **334**.

[**0050**] FIG. **12** is a flow diagram showing a process for signing up a new user of a video game with downloadable statistics. In step **390**, the CPU displays, a sign-up screen. In step **393**, the user enters the user's name and chosen password, and possibly other identifying information, such as a birthdate, address, etc. In step **396**, the process depends upon whether the AC is autosent. If the AC is autosent, the CPU sends the sign-up data **404** to the server **408** in step **400**. In step **412**, the server sends account options **416** to the game machine.

[**0051**] In step **420**, the CPU displays the account options to the user. In step **424**, the user selects what type of account the user chooses. In step **428**, the CPU sends the users choice of account type to the server. For example, the user may choose to pay for downloads of new game parameters on a monthly basis. As another example, the user may choose to pay for downloads of new game parameters for an entire sports season. Or, as yet another example, the user may choose to pay for downloads of new game parameters on a pay per use basis. Other business models will be apparent to those of skill in the art.

[**0052**] In step **432**, the server verifies the information, creates an account and sends a menu of game data to the game machine. In step **436**, the CPU displays the menu to the user. Next, the process continues as shown in FIG. **13**.

[**0053**] Alternatively, if in step **396**, the AC is not autosent, the user enters an AC, in step **440**. In step **444**, the CPU compares the user input AC with the game machine authentication code. In step **446**, the CPU determines whether the user input AC matches the game machine AC. If not, the process continues at step **448** and the CPU displays an error message to the user. If the user input AC matches the game machine AC, the process continues at step **400**.

[**0054**] FIG. **13** is a flow diagram showing a download process of a new game parameter. In step **450**, the user selects a set of desired game parameters. For example, the desired parameters may comprise a complete set of new parameters for all teams in the game. As another example, the desired parameters may comprise the results of a single sports game for a single team. In step **453**, the CPU sends the request to the server.

[**0055**] In step **456**, the process depends on whether the game machine requires an authentication code. In step **459**, the server sends the requested parameters to the CPU. In step **462**, the CPU stores the new game parameters in local memory **465**.

[**0056**] Alternatively, if in step **456**, the game machine requires an AC, the process continues to step **468**. In step **468**, the server attaches account information and the AC to the parameter. In step **471**, the server sends the parameter to the CPU. In step **474**, the CPU compares the AC with the game machine AC. In step **476**, the game machine determines whether the AC sent by the server matches the AC stored on the game machine. If there is no match, the CPU displays an error message. If there is a match, the CPU stores the data at step **479** in local memory **465**. The data, or new game parameters, can then be used for play in a game with quantifiably increased realism.

[**0057**] FIG. **14** is a flow diagram showing game play in a case in which an AC is required. In step **490**, the CPU displays a game menu. In step **495**, the user selects to load saved data **500**, which has been stored in local memory **505**. In step **510**, the CPU reads data from local memory. In step **515**, the CPU compares an AC associated with the data to the game machine AC. In step **520**, the process determines whether the AC associated with the data is the same as the game machine AC. If they don't match, the CPU displays an error message in step **535**. If they match, the CPU loads the parameter data into RAM **525** in step **530**. Then, the new game parameters are used for play in a game with quantifiably increased realism.

What is claimed is:

1.-**28**. (canceled)

29. A game medium configured to provide a sports video game in conjunction with a video game machine, the sports video game including video game rules and video game character parameters, the video game character parameters including video game character performance parameters

associated with individual video game characters, the game medium being configured to cause the video game machine to perform a method comprising:

loading video game data stored by the game medium into a random access memory of the video game machine for playing the video game, the video game data including the video game rules and a particular video game character performance parameter associated with a particular individual video game character associated with a particular real-life sports athlete, wherein the particular video game performance parameter is based at least in part on a real-life performance of the particular real-life sports athlete playing in one or more real-life sporting events, the particular video game character performance parameter affecting the manner in which the particular individual video game character performs in the sports video game;

during a single sports season, receiving a series of updated video game character performance parameters from a data server via a network including the Internet, wherein each of the updated video game character performance parameters in the series is based at least in part on one or more different real-life performances of the particular real-life sports athlete in one or more sporting events performed during the single sports season;

updating the sports video game with each of the updated video game character performance parameters received, wherein each update changes the manner in which the particular individual video game character performs in the sports video game such that the particular individual video game character more closely simulates real-life performance attributes of the particular real-life athlete in the sports video game; and

enabling a user to control the particular individual video game character in the sports video game using a video game controller connected to the video game machine.

30. A game medium as in claim 29, wherein receipt of the updated video game performance parameters in the series occurs at least weekly during at least a portion of the single sports season, such that the particular video game character simulates current real-life performance attributes of the particular real-life athlete on an at least weekly basis.

31. A game medium as in claim 29, wherein receipt of an the updated video game performance parameters in the series occurs at least daily during at least a portion of the single sports season, such that the particular video game character simulates current real-life performance attributes of the particular real-life athlete on an at least daily basis.

32. A game medium according to claim 29, the game medium being further configured to cause the video game machine to update visual aspects of the video game during a single sports season, wherein the updated visual aspects are based on a change in a real-life visual aspect of a real-life athlete, stadium, or field during a single sports season.

33. A game medium according to claim 29, the game medium being further configured to cause the video game machine to perform the method further comprising:

receiving an input from the user using the video game controller, the input directing the video game machine to connect to the data server via the internet to receive the new video game character performance parameter.

34. A game medium according to claim 29, wherein the video game enables a user to choose, or to not choose, a video game update that includes an updated video game character performance parameter.

35. A game medium according to claim 29, the game medium being further configured to cause the video game machine to perform the method periodically and automatically in response to an input received from the user using the video game controller.

36. A game medium according to claim 29, the game medium being further configured to cause the video game machine to perform the method further comprising:

receiving a change in a video game team parameter from a data server via the network, wherein the video game team parameter corresponds to a real-life trade of a real-life athlete between two real-life teams during a single sports season; and

changing a video game team in the video game based on the video game team parameter received to reflect the trade of the real-life athlete in the sports video game.

37. A game medium according to claim 29, the game medium being further configured to cause the video game machine to perform the method further comprising:

receiving an updated video game stadium parameter from a data server via the network including the Internet, wherein the video game stadium parameter corresponds to an attribute of a real-life stadium; and

changing a stadium parameter in the video game based on the updated video game stadium parameter received.

38. A game medium according to claim 37, wherein the updated video game stadium parameter is associated with an appearance of the stadium that has changed at least once during a single sports season.

39. A game medium according to claim 29, wherein the updated video game character performance parameters are each associated with a real-life performance tendency of the particular real-life athlete in the real-life performances during the single sports season.

40. A game medium according to claim 29, wherein at least one of the updated video game character performance parameters is associated with a performance of a real-life opponent of the real-life athlete in a real-life sports event during the single sports season.

41. A game medium according to claim 29, wherein at least one of the updated video game character performance parameters is associated with a points scored average of a real-life performance of the real-life athlete in a real-life sports event during the single sports season.

42. A game medium according to claim 29, wherein at least one of the updated video game character performance parameters is associated with a completion percentage of a real-life performance of the real-life athlete in a real-life sports event during a single sports season.

43. A game medium according to claim 29, wherein the sports video game is a football sports video game including video game football stadiums, video game football teams, and video game characters associated with real-life football stadiums, real-life football teams, and real-life football athletes respectively.

44. A game medium according to claim 43, wherein at least one of the updated video game character performance parameters is associated with a field goal percentage of a real-life

performance of the real-life athlete in a real-life sports event during the single sports season.

45. A game medium according to claim 43, wherein at least one of the updated video game character performance parameters is associated with rush yards of a real-life performance of the real-life athlete in a particular real-life sports event during a single sports season.

46. A game medium according to claim 29, wherein the sports video game is a basketball sports video game including video game basketball courts, video game basketball teams, and video game basketball characters associated with real-life basketball courts, real-life basketball teams, and real-life basketball athletes respectively.

47. A game medium according to claim 46, wherein at least one of the updated video game character performance parameters is associated with a free-throw completion percentage of a real-life performance of the real-life athlete in a real-life sports event during the single sports season.

48. A game medium according to claim 46, wherein at least one of the updated video game character performance parameters is associated with a shots average of a real-life performance of the real-life athlete in a real-life sports event during the single sports season.

49. A game medium according to claim 29, wherein the sports video game is a soccer sports video game including video game soccer stadiums, video game soccer teams, and video game soccer characters associated with real-life soccer stadiums, real-life soccer teams, and real-life soccer athletes respectively.

50. (canceled)

51. A method performed by a computer server for updating a video game character parameter of an individual sports video game character in a sports video game played on a video game machine during a single sports season, the sports video game character representing a particular real-life athlete in the sports video game, the method performed by the computer server comprising the following acts:

during a single sports season, storing a plurality of updated video game character performance parameters on a storage medium of the computer server, wherein each of the updated video game character performance parameters is based at least in part on a performance of the particular real-life athlete performing in one or more real-life sporting events of the single sports season;

communicating with a video game machine to establish a network connection over the Internet; and

within the single sports season, transmitting the plurality of updated video game performance parameters to the video game machine in a plurality of transmissions over the network, thereby enabling the video game machine to (i) receive the plurality of updated video game performance parameters within the single sports season, and (ii) change the sports video game play based on changes in the updated video game performance parameters, such that the particular individual video game character of the sports video game more closely simulates the real-life performance of the particular real-life athlete during the single sports season.

52. A method as in claim 51, wherein,

at least a portion of the transmissions are carried out at least weekly during at least a portion of the single sports season; and

the at least weekly transmissions include video game character performance parameters based at least in part on real-life performances of the particular real-life athlete that occurred at least within the prior week.

53. A method as in claim 51, wherein:

at least a portion of the updated transmissions are carried out at least daily during at least a portion of the single sports season; and

the at least weekly transmissions include video game character performance parameters based at least in part on real-life performances of the particular real-life athlete that occurred at least within the prior day.

54. A method as in claim 51, wherein the server is further configured to receive a request for an updated video game character performance parameter originating at the video game machine and:

verify that a payment has been made for the requested updated video game character performance parameter; or

receive authorization to execute a payment for the transfer of the updated video game character performance parameter.

55. A method as in claim 51, wherein each of the updated video game performance parameters is associated with a different real-life performance of a the particular real-life athlete during a single real-life game of the single sports season.

56. (canceled)

57. A method as in claim 51, wherein the updated video game performance parameters are each associated with the performance of:

a particular real-life basketball player;

a particular real-life football player; or

a particular real-life soccer player.

58. A method as in claim 51, wherein the particular real-life athlete is a real-life professional athlete that plays for a real-life professional team.

59. A method as in claim 51, wherein each of the updated video game character performance parameters are based at least in part on one or more different real-life sporting events against different opponents by the particular real-life athlete during the single sports season.

60. A game medium according to claim 29, wherein, wherein at least one of the plurality of updated video game character performance parameters is based on:

a current points scored average by the particular real-life athlete in a particular real-life sports event during the single sports season;

a field goal percentage of a real-life performance of the particular real-life athlete in a particular real-life sports event during the single sports season;

a rush yards of a real-life performance of the particular real-life athlete in a particular real-life sports event during a single sports season;

a free-throw completion percentage of the particular real-life athlete in a particular real-life sports event during the single sports season;

a shots average of the particular real-life athlete in a particular real-life sports event during the single sports season; or

a completion percentage of the particular real-life athlete in a particular real-life sports event during the single sports season.

61. A method as in claim 51, further comprising:

storing a second plurality of updated video game character performance parameters, each of the second plurality of updated video game character performance parameters based at least in part on performance statistics of a second particular real-life athlete performing in one or more real-life sporting events of the single sports season, each of the second updated video game character performance parameters being based at least in part on different performance statistics from the same second particular athlete and from the same single sports season; and

within the single sports season, transmitting the second plurality of updated video game character performance parameters to the video game machine in a plurality of transmissions over the network.

62.-**67**. (canceled)

68. A game medium configured to provide a basketball sports video game in conjunction with a video game machine, the basketball sports video game including video game rules and basketball video game character parameters, the basketball video game character parameters including video game character performance parameters, the game medium being configured to cause the video game machine to perform a method comprising:

loading video game data stored by the game medium into a random access memory of the video game machine for playing the basketball video game, the video game data including the video game rules and a particular basketball video game character performance parameter associated with a particular basketball video game character associated with a particular real-life basketball sports athlete, wherein the particular basketball video game performance parameter is based at least in part on a real-life basketball performance of the particular real-life basketball sports athlete playing in one or more real-life basketball sporting events, the particular basketball video game character performance parameter affecting the manner in which the particular basketball video game character performs in the basketball sports video game;

during a single basketball sports season, receiving a series of updated video game basketball character performance parameters from a data server via a network including the Internet, wherein each of the updated basketball video game character performance parameters in the series is based at least in part on one or more different real-life performances of the particular real-life basketball sports athlete performed during the single sports season;

updating the basketball sports video game with each of the updated basketball video game character performance parameters received, wherein each update changes the manner in which the particular basketball video game character performs in the basketball sports video game such that the particular basketball video game character more closely simulates real-life basketball performance attributes of the particular real-life basketball athlete in the sports video game; and

enabling a user to control the particular video game basketball character in the basketball sports video game using a video game controller connected to the basketball video game machine, wherein:

the basketball sports video game includes video game basketball courts, video game basketball teams, and video game basketball characters associated with real-life basketball courts, real-life basketball teams, and real-life basketball athletes respectively; and

at least one of the updated video game character performance parameters is associated with a free-throw completion percentage of a real-life performance of the real-life athlete in a real-life sports event during the single sports season.

69. A game medium according to claim 68, wherein at least one of the updated video game character performance parameters is associated with a shots average of a real-life performance of the real-life athlete in a real-life basketball sports event during the single basketball sports season.

70. A game medium as in claim 68, wherein receipt of the updated basketball video game performance parameters in the series occurs at least weekly during at least a portion of the single basketball sports season, such that the particular basketball video game character simulates current real-life performance attributes of the particular real-life basketball athlete on an at least weekly basis.

71. A game medium as in claim 68, wherein receipt of an the updated basketball video game performance parameters in the series occurs at least daily during at least a portion of the single basketball sports season, such that the particular video game character simulates current real-life performance attributes of the particular real-life basketball athlete on an at least daily basis.

* * * * *